



**T.C  
FIRAT ÜNİVERSİTESİ  
MÜHENDİSLİK FAKÜLTESİ  
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

**İNTERNET SERVİS SAĞLAYICI OTOMASYONU  
PROJESİ**

**RAPOR**

**ARİF DAĞ 220260030  
YUSUF SOYSAL 220260034  
ÇAĞRI AYDIN 220260050**

# 1. Proje Konusu

Bu projede bir **İnternet Servis Sağlayıcı (ISP) Otomasyon Sistemi** geliştirilecektir. Bu sistem, müşterilerin internet hizmetlerine kolayca abone olmasını, ödeme yapmasını, teknik destek taleplerini yönetmesini ve geçmiş ödemelerini görüntülemesini sağlar ve ISP personelinin ve yöneticilerinin de müşteri ve abonelik işlemlerini düzenleyip izlemelerine olanak tanır.

## 2. Kullanıcı Roller ve Gereksinimleri

Projede farklı roller için farklı gereksinimler olacaktır:

Projede kullanıcı rolleri ve roller için yetki ve gereksinimler aşağıdaki şekilde belirlenmiştir:

### 1. Müşteri Rolü:

- Paket seçebilir, bir paket aboneliği alabilir veya paket değiştirebilir.
- Mevcut abonelikleri görüntüleyip inceleyebilir.
- Ödemelerini ve geçmiş ödeme kayıtlarını görebilir.
- Teknik destek talebi oluşturabilir, durumunu takip edebilir ve geçmiş taleplerini de görüntüleyebilir.

### 2. Teknik Destek Personeli:

- Bütün müşterilerin teknik destek taleplerini görüp yanıt verebilir.
- Destek taleplerine dair durum güncellemeleri yapabilir ve gerekirse talepleri kapatabilir.
- Atanmış cihazların durumunu güncelleyebilir ve müşterilere cihaz ekleyebilir veya değiştirebilir.
- Sistemdeki logları görüntüleyebilir, hatalı işlemleri analiz edebilir.

### 3. Mağaza Personeli:

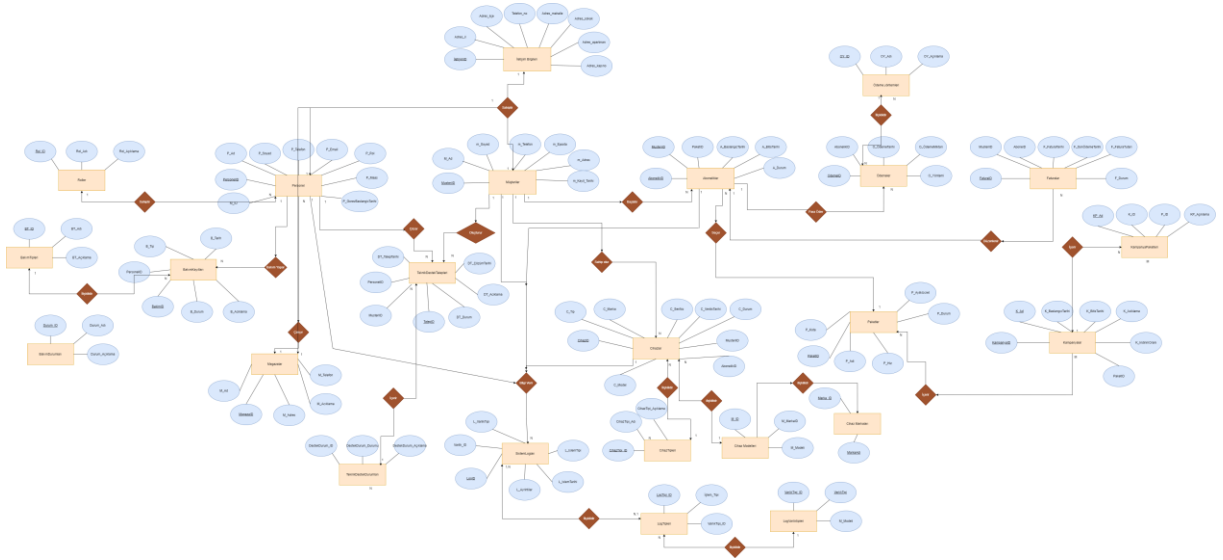
- Hem mağaza kayıtlarını düzenleyebilir hem de müşterilerle ilgili güncelleme yapabilir.
- Mağazada yapılan bakım işlemlerini kayıt altına alabilir ve bakım geçmişini görüntüleyebilir.

### 4. Yönetici Rolü:

- Sistemdeki bütün müşterilere, aboneliklere, ödemelere ve destek taleplerine erişebilir ve düzenleyebilir.
- Yeni paketler tanımlayabilir, var olan paketlerin özelliklerini güncelleyebilir ve kampanya ekleyebilir.
- Yeni mağaza ve personel kayıtları ekleyebilir, mevcut kayıtları düzenleyebilir.
- Sistem loglarına ve bakım kayıtlarına erişerek gerekli analizleri yapabilir.

## 3. E-R Diyagramı

Projenin E-R diyagramı:



Daha iyi çözünürlüklü versiyonuna [linkten](#) veya proje sayfasından ulaşılabilir.

## 4. Varlıkların Nitelikleri, İlişkileri ve Kısıtlamaları

### 1. Müşteriler Tablosu

- **Gereksinimler:** Bir müşteri, birden fazla abonelik başlatabilir ve teknik destek talebinde bulunabilir.
- **İlişkiler:**
  - Abonelikler tablosu ile 1:N (Bir müşteri birden fazla abonelik oluşturabilir, ama bir abonelik sadece bir müşteriyle ilişkilidir).
  - Teknik Destek Talepleri tablosu ile 1:N (Bir müşteri birçok destek talebi açabilir).
  - Sistem Logları tablosu ile 1:N.
  - İletişim Bilgileri Tablosu ile 1:1
- **Kısıtlamalar:**
  - MusteriID: Her müşterinin bir kimlik numarası olmalı (Primary Key).
  - Ad: Müşteri adı, boş olamaz.
  - Soyad: Müşteri soyadı, boş olamaz.
  - İletişim\_ID: Foreign Key, **İletişim Bilgileri** tablosuna referans.
  - KayıtTarihi: Kayıt tarihi. Boş olamaz

### 2. İletişim Bilgileri Tablosu

- **Gereksinimler:** İletişim bilgilerini tutar.
- **İlişkiler:**
  - Müşteriler tablosu ile 1:1
  - Mağazalar tablosu ile 1:1
  - Personel tablosu ile 1:1

- **Kısıtlamalar:**
  - İletisimID: Her İletişim bilgisi bir kimlik numarasına sahip olmalı (Primary Key).
  - Telefon: Telefon numarası, boş olamaz(UNIQUE).
  - Email: E-posta adresi, boş olamaz(UNIQUE).
  - İl: Adresin ili.
  - İlçe: Adresin ilçesi.
  - Mahalle: Adresin mahallesi.
  - Sokak: Adresin sokağı.
  - Apartman: Apartman bilgisi.
  - KapıNo: Kapı numarası.

### 3. Paketler Tablosu

- **Gereksinimler:** Hız ve kota ile ilgili farklı özellikleri olan internet paketleri tanımlanabilir. Paketlerin aktif ya da pasif durumu olmalıdır.
- **İlişkiler:**
  - Abonelikler tablosu ile 1:N (Bir paket, birden fazla abonelikte seçilebilir, ama her abonelik sadece bir paketle ilişkilidir).
  - Kampanya Paketleri tablosu ile 1:N.
- **Kısıtlamalar:**
  - PaketID: Her paketin bir paket numarası olmalı (Primary Key).
  - PaketAdi: Paket adı, boş olamaz.
  - Hız: İnternet hızı (Mbps). Pozitif tam sayı olmalıdır.
  - Kota: GB cinsinden kota. Pozitif tam sayı olmalıdır.
  - AylıkÜcret: Paket ücreti. Pozitif tam sayı olmalıdır.
  - Durum: Paketin aktif/pasif durumu. Yalnızca aktif veya pasif olabilir.(1 veya 0 olabilir)

### 4. Abonelikler Tablosu

- **Gereksinimler:** Her müşterinin, seçtiği pakete göre aboneliği olmalı. Aboneliklerin başlangıç ve bitiş tarihleri bulunmalı.
- **İlişkiler:**
  - Müşteriler tablosu ile N:1.
  - Paketler tablosu ile N:1.
  - Ödemeler tablosu ile 1:N (Bir aboneliğe birden fazla ödeme yapılabilir).
  - Faturalar tablosu ile 1:N.
  - Sistem Logları tablosu ile 1:N.
- **Kısıtlamalar:**
  - AbonelikID: Her abonelik için bir kimlik numarası olmalı (Primary Key).
  - MusteriID: Her abonelik bir müşteri kimlik numarasına sahip olmalı (Foreign Key). **Müşteriler** tablosuna referans.
  - PaketID: Her abonelik bir paket kimlik numarasına sahip olmalı. (Foreign Key). **Paketler** tablosuna referans.
  - BaslangicTarihi: Abonelik başlangıç tarihi. Boş bırakılmaz.
  - BitisTarihi: Abonelik bitiş tarihi.

- **Durum:** Aboneliğin aktif/pasif durumu. Yalnızca aktif (1) veya pasif (0) olabilir.

## 5. Ödemeler Tablosu

- **Gereksinimler:** Aboneliklere yönelik yapılan ödemeler ve ödeme yöntemleri kaydedilmeli.
- **İlişkiler:**
  - Abonelikler tablosu ile N:1.
  - Faturalar tablosu ile 1:1.
  - Ödeme Yöntemleri tablosu ile N:1.
- **Kısıtlamalar:**
  - **OdemeID:** Her ödeme için bir kimlik numarası olmalı (Primary Key).
  - **AbonelikID:** Her ödeme için bir abone kimlik numarası olmalı (Foreign Key). **Abonelikler** tablosuna referans.
  - **OdemeYontemiID:** (kredi kartı vb.) boş bırakılamaz. **Odeme Yontemleri** tablosuna referans (Foreign Key).
  - **OdemeTarihi:** Ödeme tarihi.
  - **OdemeMiktari:** Pozitif değer olmalı.
  - Aynı tarihte tekrarlayan ödemeleri engellemek için (AbonelikID – OdemeTarihi) birleşik kısıtlaması kullanılacak.

## 6. Ödeme Yöntemleri Tablosu

- **Gereksinimler:** Ödeme yöntemlerini (örneğin, "Kredi Kartı", "Banka Transferi") tanımlar.
- **İlişkiler:**
  - Ödemeler tablosu ile 1:N.
- **Kısıtlamalar:**
  - **OdemeYontemiID:** Her ödeme yöntemi için bir kimlik numarası olmalı (Primary Key).
  - **OdemeyeYontemiAdi:** Ödeme yöntemi adı (örneğin, "Kredi Kartı"). Eşsiz olmalı ve boş bırakılamaz.
  - **Aciklama:** Ödeme yöntemi hakkında açıklama.

## 7. Faturalar Tablosu

- **Gereksinimler:** Aboneliklere belirli bir tarihte oluşturulan faturalar tutulmalı.
- **İlişkiler:**
  - Abonelikler tablosu ile 1:N.
  - Faturalar tablosu ile 1:1.
- **Kısıtlamalar:**
  - **FaturaID:** Her fatura için bir kimlik numarası olmalı (Primary Key).
  - **AbonelikID:** Her fatura için bir abonelik kimlik numarası olmalı (Foreign Key). **Abonelikler** tablosuna referans.
  - **FaturaTarihi:** Faturanın düzenlendiği tarih.
  - **SonOdemeTarihi:** Faturanın son ödeme tarihi. Boş olmaz.
  - **FaturaTutari:** Fatura toplam tutarı. Pozitif olmalıdır.
  - **Durum:** Faturanın ödenip ödenmediğini belirten durum. Yalnızca ödenmiş veya ödenmemiş olabilir.(Ödenmişse 1, ödenmemişse 0)

- **Ödemeler** ve **Fatura** .Durum arasındaki tutarlılık için trigger eklenecek.

## 8. Teknik Destek Talepleri Tablosu

- **Gereksinimler:** Müşterilerin destek talebi oluşturabilmesi ve talep durumlarının takip edilmesi gerekir.
- **İlişkiler:**
  - Müşteriler tablosu ile N:1.
  - Personel tablosu ile 1:N.
  - Teknik Destek Durumları tablosu ile N:1.
- **Kısıtlamalar:**
  - DestekTalepID: Her destek talebi için bir kimlik numarası olmalı (Primary Key).
  - MusteriID: Her fatura bir müşteri kimlik numarasına sahip olmalı (Foreign Key). **Müşteriler** tablosuna referans.
  - PersonelID: **Personel** tablosuna referans (isteğe bağlı, müşteri talep açtıktan sonra teknik destek personeline atanabilir) (Foreign Key).
  - TalepTarihi: Teknik destek talebinin oluşturulma tarihi.
  - DurumID: Her teknik destek talebi bir tane durum kimlik numarasına sahip olmalı (Foreign Key). **Teknik Destek Durumları** tablosuna referans
  - Aciklama: Talep hakkında açıklama.
  - CozumTarihi: Talep çözüldüğünde doldurulacak tarih.

## 9. Teknik Destek Durumları Tablosu

- **Gereksinimler:** Sistemdeki teknik destek durumlarını ("Beklemede " ya da "Çözüldü" vb.) tanımlar.
- **İlişkiler:**
  - Teknik Destek Talepleri tablosu ile 1:N.
- **Kısıtlamalar:**
  - DestekDurumID: : Her destek durumu için bir kimlik numarası olmalı (Primary Key).
  - Durum: Durum adı, boş olamaz.
  - Aciklama: Durum hakkında ek açıklama.

## 10. Personel Tablosu

- **Gereksinimler:** Personelin adı, pozisyonu, iletişim bilgileri tutulmalı.
- **İlişkiler:**
  - Teknik Destek Talepleri tablosu ile 1:N.
  - Bakım Kayıtları tablosu ile 1:N.
  - Mağazalar tablosu ile N:1.
  - Sistem Logları tablosu ile 1:N.
  - İletişim Bilgileri Tablosu ile 1:1
  - Roller tablosu ile 1:N.

- **Kısıtlamalar:**
  - PersonelID: Her personel için bir kimlik numarası olmalı (Primary Key).
  - MağazaID: Foreign Key, **Mağazalar** tablosuna referans (sadece mağaza personeli için dolu olacak).
  - Ad: Personel adı. Boş olamaz.
  - Soyad: Personel soyadı. Boş olamaz.
  - İletişim\_ID: Foreign Key, **İletişim Bilgileri** tablosuna referans.
  - RolID: Her personel bir rol kimlik numarasına sahip olmalı (Foreign Key). **Roller** tablosuna referans.
  - GorevBaslangicTarihi: Personelin göreve başlama tarihi.
  - Maas: pozitif olmalı.

## 11. Roller Tablosu

- **Gereksinimler:** Sistemdeki kullanıcı rollerini (örneğin, "Teknik Destek", "Yönetici") tanımlar.
- **İlişkiler:**
  - Personel tablosu ile 1:N.
- **Kısıtlamalar:**
  - RolID: : Her rol bir kimlik numarasına sahip olmalı (Primary Key).
  - RolAdi: Rol adı, boş olamaz.
  - Aciklama: Rol hakkında ek açıklama.

## 12. Mağazalar Tablosu

- **Gereksinimler:** Mağaza bilgileri ve her mağazanın adresi ve iletişim numarası kaydedilmeli.
- **İlişkiler:**
  - Personel tablosu ile 1:N.
  - İletişim Bilgileri Tablosu ile 1:1
- **Kısıtlamalar:**
  - MağazaID: : Her mağazanın bir kimlik numarası olmalı (Primary Key).
  - MağazaAdi: Mağaza adı, boş olamaz.
  - İletişim\_ID: Foreign Key, **İletişim Bilgileri** tablosuna referans.
  - Aciklama: Mağaza hakkında ek bilgi.

## 13. Cihazlar Tablosu

- **Gereksinimler:** Müşteriye ait olabilecek cihaz bilgileri (modem, router vb.) tutulmalı.

- **İlişkiler:**
  - Müşteriler tablosu ile N:1 (Bir müşterinin birden fazla cihazı olabilir, her cihaz sadece tek bir müşteriye atanabilir).
  - Sistem Logları tablosu ile 1:N.
  - Cihaz Tipleri tablosu ile N:1.
  - Cihaz Modelleri tablosu ile N:1.
- **Kısıtlamalar:**
  - CihazID: Her cihazın bir kimlik numarası olmalı (Primary Key).
  - MusteriID: Her cihazın bir müşteri kimlik numarası olmalı (Foreign Key). **Müşteriler** tablosuna referans.
  - AbonelikID: Her cihazın bir abonelik kimlik numarası olmalı (Foreign Key). **Aboneler** tablosuna referans (Null olabilir).
  - CihazTipiID: Her cihazın bir cihaz tipi kimlik numarası olmalı (Foreign Key). **Cihaz Tipi** tablosuna referans (Null olabilir).
  - ModelID: Her cihazın bir model kimlik numarası olmalı (Foreign Key). **Cihaz Model** tablosuna referans (Null olabilir).
  - SeriNumarasi: Cihazın seri numarası, benzersiz olmalı ve boş olamaz.
  - VerilisTarihi: Cihazın müşteriye verildiği tarih.
  - Durum: Cihazın aktif/pasif durumu. Durum aktif (1) veya pasif (0) olmalıdır.

## 14. Cihaz Tipleri Tablosu

- **Gereksinimler:** Cihaz türlerini (Örneğin, “modern”, “router”) tutar.
- **İlişkiler:**
  - Cihazlar tablosu ile 1:N.
- **Kısıtlamalar:**
  - CihazTipiID: Her cihaz tipi için bir kimlik numarası olmalı (Primary Key).
  - CihazTipiAdi: Cihazın adı (modem, router vb.), boş olamaz.
  - Aciklama: Cihaz hakkında ek açıklama.

## 15. Cihaz Modelleri Tablosu

- **Gereksinimler:** Cihaz model bilgilerini tutar.
- **İlişkiler:**
  - Cihazlar tablosu ile 1:N.
  - Cihaz markaları tablosu ile N:1.
- **Kısıtlamalar:**
  - ModelID: Her cihaz modeli için bir kimlik numarası olmalı (Primary Key).
  - MarkaID: Her cihaz modeli için bir cihaz markası kimlik numarası olmalı (Foreign Key). **Cihaz Markaları** tablosuna referans.
  - Model: Cihaz modeli.



## 16. Cihaz Markaları Tablosu

- **Gereksinimler:** Cihaz marka bilgilerini tutar.
- **İlişkiler:**
  - Cihaz modelleri tablosu ile 1:N.
- **Kısıtlamalar:**
  - MarkaID: Her cihaz markasının bir kimlik numarası olmalı (Primary Key).
  - Marka: Cihaz markası.

## 17. Kampanyalar Tablosu

- **Gereksinimler:** Kampanya bilgileri, indirim oranları ve tarih aralığına göre kaydedilmeli.
- **İlişkiler:**
  - Kampanya Paketleri tablosu ile 1:N.
- **Kısıtlamalar:**
  - KampanyaID: Her kampanya bir kimlik numarasına sahip olmalı (Primary Key).
  - KampanyaAdi: Kampanya adı, boş olamaz.
  - BaslangicTarihi: Kampanyanın başlangıç tarihi
  - BitisTarihi: Kampanyanın bitiş tarihi.
  - IndirimOrani: Kampanya kapsamında sunulan indirim oranı (% cinsinden). %0 ile %100 arasında bir değer olmalı.
  - Aciklama: Kampanya hakkında açıklama.

## 18. Kampanya Paketleri Tablosu

- **Gereksinimler** Kampanyalar ve paketler arasındaki ilişkileri tutar.
- **İlişkiler:**
  - Kampanyalar tablosu ile N:1.
  - Paketler tablosu ile N:1.
- **Kısıtlamalar:**
  - KampanyaPaketID: Her kampanya paket ilişkisi bir kimlik numarasına sahip olmalı (Primary Key).
  - KampanyaID: Her kampanya paketi bir kampanya numarasına sahip olmalı (Foreign Key). **Kampanya** tablosuna referans.
  - PaketID: Her kampanya paket ilişkisinin bir paket numarası olmalı (Foreign Key). **Paket** tablosuna referans.
  - EkAçiklama: İsteğe bağlı alan, paketlerin kampanya içindeki durumu açıklanabilir

## 19. Sistem Logları Tablosu

- **Gereksinimler:** Sistem hareketlerini izlemek için log kayıtları tutulmalı.

- **İlişkiler:**
  - Müşteriler tablosu ile 1:N.
  - Abonelikler tablosu ile 1:N.
  - Personel tablosu ile 1:N.
  - Cihazlar tablosu ile 1:N.
  - Log Tipleri tablosu ile N:1.
- **Kısıtlamalar:**
  - LogID: Her log için kimlik numarası olmalı (Primary Key).
  - VarlıkID: Log kaydının ilişkilendirildiği işlemi yapan varlığın kimlik numarası(Foreign Key). Mesela eğer işlem bir **Cihaz** ile ilgiliyse buraya **CihazID** gelecek. Boş olamaz.
  - LogTipiID: Her sistem logu bir log tipi numarasına sahip olmalı (Foreign Key).**Log Tipleri** tablosuna referans (bakımı yapan kişi).
  - IslemTarihi: Log kaydının oluşturulduğu tarih ve saat, boş olamaz.
  - Ayrıntılar: İşlemle ilgili açıklama veya ayrıntılar içeren bir alan.

## 20. Log Tipleri Tablosu

- **Gereksinimler:** Sistemdeki log tiplerini tutar.
- **İlişkiler:**
  - Sistem Logları tablosu ile 1:N.
  - Sistem Logları tablosu ile N:1.
- **Kısıtlamalar:**
  - LogTipiID: Her log tipinin kimlik numarası olmalı (Primary Key).
  - İşlemTipi: Logun tipini belirten bir alan. Mesela, "Oluşturma", "Güncelleme", "Silme", "Giriş", "Çıkış" gibi işlemler tanımlarak tip belirlenir. Boş Olamaz.
  - VarlıkTipiID: Her log tipi için bir log varlık tipi numarası olmalı (Foreign Key).**Log Varlık Tipleri** tablosuna referans.

## 21. Log Varlık Tipleri Tablosu

- **Gereksinimler:** Sistemdeki log varlık tiplerini (örneğin, "Cihaz ", "Müşteri") tanımlar.
- **İlişkiler:**
  - Log Tipleri tablosu ile 1:N.
- **Kısıtlamalar:**
  - VarlıkTipiID: : Her varlık tipi bir kimlik numarasına sahip olmalı (Primary Key).
  - VarlıkTipi: Hangi tabloyla ilişkili olduğunu belirten alan. Mesela, Cihaz, Müşteri, Personel gibi değerler tutarak logun hangi varlıkla alakalı olduğunu belirtilir. Boş olamaz.
  - Açıklama: Varlık tipi hakkında ek açıklama.

## 22. Bakım Kayıtları Tablosu

- **Gereksinimler:** Mağazalarda yapılan bakım işlemleri takip edilmelidir.
- **İlişkiler:**
  - Personel tablosu ile N:1.
  - Bakım Tipleri tablosu ile N:1.
  - Bakım Tipleri tablosu ile N:1.
- **Kısıtlamalar:**
  - BakimID: Her bakım kaydının kimlik numarası olmalı (Primary Key).
  - PersonelID: Her bakım kaydının bir personel numarası olmalı (Foreign Key). **Personel** tablosuna referans (bakımı yapan kişi).
  - BakimTipiID: Her bakım kaydı bir bakım tipi numarasına sahip olmalı (Foreign Key). **Bakım Tipleri** tablosuna referans (bakımı yapan kişi).
  - BakimTarihi: Bakımın yapıldığı tarih, boş olamaz.
  - Aciklama: Bakım hakkında detaylı açıklama.
  - DurumID: Her bakım kaydı bir durum numarasına sahip olmalı (Foreign Key). **Bakım Durumları** tablosuna referans.

## 23. Bakım Tipleri Tablosu

- **Gereksinimler:** Sistemdeki bakım tiplerini (örneğin, “Yazılım Güncellemesi”, “Donanım Bakımı”) tutar.
- **İlişkiler:**
  - Bakım Kayıtları tablosu ile 1:N.
- **Kısıtlamalar:**

BakimTipiID: Her bakım tipinin kimlik numarası olmalı (Primary Key).

- BakimTipiAdi: Bakım tipinin adı. (Boş Olamaz)
- Aciklama: Bakım tipi hakkında aciklama.

## 24. Bakım Durumları Tablosu

- **Gereksinimler:** Sistemdeki bakım durumlarını (örneğin, “Tamamlandı”, “Beklemede”) tutar.
- **İlişkiler:**
  - Bakım Kayıtları tablosu ile 1:N.
- **Kısıtlamalar:**
  - DurumID: Her bakım durumu kimlik numarasına sahip olmalı (Primary Key).
  - DurumAdi: Bakımın durumu (tamamlandı, beklemede vb.).
  - Aciklama: Bakım durumu hakkında aciklama.

## 5. İlişki Şeması

1) Müşteriler(MusteriID, Ad, Soyad, Iletisim\_ID, KayitTarihi)

2) IletisimBilgileri(IletisimID, Telefon, Email, Il, Ilce, Mahalle, Sokak, Apartman, KapiNo)

3) Paketler(PaketID, PaketAdi, Hiz, Kota, AylikUcret, Durum)

- 4) Abonelikler(AbonelikID, MusteriID, PaketID, BaslangicTarihi, BitisTarihi, Durum)
- 5) Odemeler(OdemeID, AbonelikID, OdemeYontemiID, OdemeTarihi, OdemeMiktari)
- 6) OdemeYontemleri(OdemeYontemiID, OdemeYontemiAdi)
- 7) Faturalar(FaturaID, AbonelikID, FaturaTarihi, SonOdemeTarihi, FaturaTutari, Durum)
- 8) TeknikDestekTalepleri(DestekTalepID, MusteriID, PersonelID, TalepTarihi, DurumID, Aciklama, CozumTarihi)
- 9) TeknikDestekDurumları(DestekDurumID, Durum, Aciklama)
- 10) Personel(PersonelID, MagazaID, Ad, Soyad, Iletisim\_ID, RolID, GorevBaslangicTarihi, Maas)
- 11) Roller(RolID, RolAdi, Aciklama)
- 12) Magazalar(MagazaID, MagazaAdi, Iletisim\_ID, Aciklama)
- 13) Cihazlar(CihazID, MusteriID, AbonelikID, CihazTipiID, ModelID, SeriNumarasi, VerilisTarihi, Durum)
- 14) CihazTipi(CihazTipiID, CihazTipiAdi, Aciklama)
- 15) CihazModelleri(ModelID, MarkaID, Model)
- 16) CihazMarkaları(MarkaID, Marka)
- 17) Kampanyalar(KampanyaID, KampanyaAdi, BaslangicTarihi, BitisTarihi, IndirimOrani, Aciklama)
- 18) KampanyaPaketleri(KampanyaPaketID, KampanyaID, PaketID, EkAciklama)
- 19) SistemLoglari(LogID, VarlikID, LogTipiID, IslemTarihi, Ayrintilar)
- 20) LogTipleri(LogTipiID, IslemTipi, VarlikTipiID)
- 21) LogVarlıkTipleri(VarlikTipiID, VarlikTipi, Aciklama)
- 22) BakimKayitlari(BakimID, PersonelID, BakimTipiID, BakimTarihi, Aciklama, DurumID)
- 23) BakimTipleri(BakimTipiID, BakimTipiAdi, Aciklama)
- 24) BakimDurumları(DurumID, DurumAdi, Aciklama)

## 6. Tablo Güncelleme ve Normalizasyon

### 1. Cihazlar Tablosu

“Marka” ve “Model” arasında geçişli bağımlılık olduğundan bunlar normalize edildi ve “Markalar” ve “Modeller” tabloları oluşturuldu.

### 2. Ödemeler Tablosu

“OdemeYontemi” alanı tekrarlı değerler içerdiğinden **Odeme Yontemleri** isminde yeni tablo oluşturuldu. **Ödemeler** tablosundaki “OdemeYontemi” kısmına “OdemeYontemiID” yazıldı.

### 3. Personel Tablosu

“Rol” alanı tekrarlı değerler içerdiğinden **Roller** isminde yeni tablo oluşturuldu. **Personel** tablosunda “Rol” alanına “RollID” yazıldı.

### 4. Personel, Mağaza, Müşteri Tabloları

Bu tablolaradaki “Telefon”, “Email” ve “Adres” bilgileri tekrarlı değerler içerdiğinden bu bilgiler **İletişim Bilgileri** tablosuna taşındı. İlgili tablolarada bu alanlar yerine “İletişimBilgisiID” yazıldı.

### 5. Bakım Kayıtları Tablosu

“BakimTipi” alanı tekrar eden değerler içerdiği için **Bakim Tipleri** isminde yeni bir tablo oluşturuldu. **Bakım Kayıtları** tablosundaki “BakimTipi” kısmına “BakimTipiID” yazıldı.

### 6. Cihazlar Tablosu

“CihazTipi” alanı tekrarlı değerler içerdiği için **Cihaz Tipleri** isminde yeni bir tablo oluşturuldu. **Cihazlar** tablosundaki “CihazTipi” kısmına “CihazTipiID” yazıldı.

### 7. Sistem Logları Tablosu

“VarlıkTipi” ve “İslemTipi” alanları tekrarlı değerler içerdiğinden sırasıyla **VarlıkTipleri** ve **İslemTipleri** isminde yeni tablolar oluşturuldu. **Sistem Logları** tablosundaki bu alanlara karşılık gelen “VarlıkTipiID” ve “İslemTipiID” kullanıldı.

### 8. Teknik Destek Talepleri Tablosu:

“Durum” alanı tekrarlı değerler içerdiğinden **Durumlar** isminde yeni bir tablo oluşturuldu. **Teknik Destek Talepleri** tablosundaki “Durum” kısmına “DurumID” yazıldı.

### 9. Log Tipleri Tablosu:

“VarlıkTipi” alanı tekrarlı değerler içerdiğinden bu bilgi **VarlıkTipleri** tablosuna taşındı. **Log Tipleri** tablosundaki ilgili alana karşılık “VarlıkTipiID” yazıldı.

### 10. Bakım Kayıtları Tablosu:

“Durum” alanı tekrarlı değerler içerdiğinden bu bilgi **Durumlar** adlı tabloya taşındı. **Bakım Kayıtları** tablosundaki ilgili alana karşılık “DurumID” yazıldı.

# 7. Kodlar

## 1) Veritabanı Oluşturma

```
create database InternetServisSaglayici
```

## 2) Tabloları Oluşturma

### 1. İletişim Bilgileri Tablosu:

```
CREATE TABLE IletisimBilgileri (  
    IletisimID INT IDENTITY(1,1) PRIMARY KEY,  
    Telefon NVARCHAR(15) NOT NULL UNIQUE,  
    Email NVARCHAR(100) NOT NULL UNIQUE,  
    Il NVARCHAR(50),  
    Ilce NVARCHAR(50),  
    Mahalle NVARCHAR(50),  
    Sokak NVARCHAR(50),  
    Apartman NVARCHAR(50),  
    KapiNo NVARCHAR(10)  
);
```

### 2. Müşteriler Tablosu:

```
CREATE TABLE Musteriler (  
    MusteriID INT IDENTITY(1,1) PRIMARY KEY,  
    Ad NVARCHAR(50) NOT NULL,  
    Soyad NVARCHAR(50) NOT NULL,  
    Iletisim_ID INT NOT NULL,  
    KayitTarihi DATE NOT NULL,  
    CONSTRAINT FK_Musteriler_Iletisim FOREIGN KEY (Iletisim_ID) REFERENCES  
    IletisimBilgileri(IletisimID)  
);
```

### 3. Paketler Tablosu:

```
CREATE TABLE Paketler (  
    PaketID INT IDENTITY(1,1) PRIMARY KEY,  
    PaketAdi NVARCHAR(100) NOT NULL,  
    Hiz INT CHECK (Hiz > 0),  
    Kota INT CHECK (Kota > 0),  
    AylikUcret DECIMAL(10, 2) CHECK (AylikUcret > 0),  
    Durum BIT NOT NULL CHECK (Durum IN (0, 1))  
);
```

### 4. Abonelikler Tablosu:

```
CREATE TABLE Abonelikler (  
    AbonelikID INT IDENTITY(1,1) PRIMARY KEY,  
    MusteriID INT NOT NULL,  
    PaketID INT NOT NULL,  
    BaslangicTarihi DATE NOT NULL,  
    BitisTarihi DATE,  
    Durum BIT NOT NULL CHECK (Durum IN (0, 1)),  
    CONSTRAINT FK_Abonelikler_Musteriler FOREIGN KEY (MusteriID) REFERENCES  
    Musteriler(MusteriID),  
    CONSTRAINT FK_Abonelikler_Paketler FOREIGN KEY (PaketID) REFERENCES  
    Paketler(PaketID)  
);
```

### 5. Ödeme Yöntemleri Tablosu:

```
CREATE TABLE OdemeYontemleri (  
    -- Table structure would follow here
```

```

OdemeYontemiID INT IDENTITY(1,1) PRIMARY KEY,
OdemeYontemiAdi NVARCHAR(50) NOT NULL UNIQUE,
Aciklama NVARCHAR(MAX)
);

```

#### 6. Ödemeler Tablosu:

```

CREATE TABLE Odemeler (
    OdemeID INT IDENTITY(1,1) PRIMARY KEY,
    AbonelikID INT NOT NULL,
    OdemeYontemiID INT NOT NULL,
    OdemeTarihi DATE NOT NULL,
    OdemeMiktari DECIMAL(10, 2) NOT NULL CHECK (OdemeMiktari > 0),
    CONSTRAINT UQ_Odemeler UNIQUE (AbonelikID, OdemeTarihi),
    CONSTRAINT FK_Odemeler_Abonelikler FOREIGN KEY (AbonelikID) REFERENCES
Abonelikler(AbonelikID),
    CONSTRAINT FK_Odemeler_OdemeYontemleri FOREIGN KEY (OdemeYontemiID) REFERENCES
OdemeYontemleri(OdemeYontemiID)
);

```

#### 7. Faturalar Tablosu:

```

CREATE TABLE Faturalar (
    FaturaID INT IDENTITY(1,1) PRIMARY KEY,
    AbonelikID INT NOT NULL,
    FaturaTarihi DATE NOT NULL,
    SonOdemeTarihi DATE NOT NULL,
    CONSTRAINT CK_SonOdemeTarihi CHECK (SonOdemeTarihi >= FaturaTarihi),
    FaturaTutari DECIMAL(10, 2) NOT NULL CHECK (FaturaTutari > 0),
    Durum BIT NOT NULL CHECK (Durum IN (0, 1)),
    CONSTRAINT FK_Faturalar_Abonelikler FOREIGN KEY (AbonelikID) REFERENCES
Abonelikler(AbonelikID)
);

```

#### 8. Magazalar Tablosu:

```

CREATE TABLE Magazalar (
    MagazaID INT IDENTITY(1,1) PRIMARY KEY,
    MagazaAdi NVARCHAR(100) NOT NULL,
    Iletisim_ID INT NOT NULL,
    Aciklama NVARCHAR(MAX),
    CONSTRAINT FK_Magazalar_Iletisim FOREIGN KEY (Iletisim_ID) REFERENCES
IletisimBilgileri(IletisimID)
);

```

#### 9. Roller Tablosu:

```

CREATE TABLE Roller (
    RolID INT IDENTITY(1,1) PRIMARY KEY,
    RolAdi NVARCHAR(50) NOT NULL,
    Aciklama NVARCHAR(MAX)
);

```

#### 10. Personel Tablosu:

```

CREATE TABLE Personel (
    PersonelID INT IDENTITY(1,1) PRIMARY KEY,
    MagazaID INT,
    Ad NVARCHAR(50) NOT NULL,
    Soyad NVARCHAR(50) NOT NULL,
    Iletisim_ID INT NOT NULL,
    RolID INT NOT NULL,
    GorevBaslangicTarihi DATE NOT NULL,
    Maas DECIMAL(10, 2) NOT NULL CHECK (Maas > 0),
    CONSTRAINT FK_Personel_Iletisim FOREIGN KEY (Iletisim_ID) REFERENCES
IletisimBilgileri(IletisimID),
    CONSTRAINT FK_Personel_Roller FOREIGN KEY (RolID) REFERENCES Roller(RolID),
);

```

```

        CONSTRAINT FK_Personel_Magazalar FOREIGN KEY (MagazaID) REFERENCES
Magazalar(MagazaID)
);

```

#### 11. Teknik Destek Durumları Tablosu:

```

CREATE TABLE TeknikDestekDurumlari (
    DestekDurumID INT IDENTITY(1,1) PRIMARY KEY,
    Durum NVARCHAR(50) NOT NULL UNIQUE,
    Aciklama NVARCHAR(MAX)
);

```

#### 12. Teknik Destek Talepleri Tablosu:

```

CREATE TABLE TeknikDestekTalepleri (
    DestekTalepID INT IDENTITY(1,1) PRIMARY KEY,
    MusteriID INT NOT NULL,
    PersonelID INT,
    TalepTarihi DATE NOT NULL,
    DurumID INT NOT NULL,
    Aciklama NVARCHAR(MAX),
    CozumTarihi DATE,
    CONSTRAINT FK_TeknikDestek_Musteriler FOREIGN KEY (MusteriID) REFERENCES
Musteriler(MusteriID),
    CONSTRAINT FK_TeknikDestek_Personel FOREIGN KEY (PersonelID) REFERENCES
Personel(PersonelID),
    CONSTRAINT FK_TeknikDestekDurum FOREIGN KEY (DurumID) REFERENCES
TeknikDestekDurumlari(DestekDurumID)
);

```

#### 13. Cihaz Tipleri Tablosu:

```

CREATE TABLE CihazTipleri (
    CihazTipiID INT IDENTITY(1,1) PRIMARY KEY,
    CihazTipiAdi NVARCHAR(50) NOT NULL,
    Aciklama NVARCHAR(255)
);

```

#### 14. Cihaz Modelleri Tablosu:

```

CREATE TABLE CihazModelleri (
    ModelID INT IDENTITY(1,1) PRIMARY KEY,
    MarkaID INT NOT NULL,
    Model NVARCHAR(50) NOT NULL,
    CONSTRAINT FK_CihazModelleri_CihazMarkalari
        FOREIGN KEY (MarkaID) REFERENCES CihazMarkalari(MarkaID),
    CONSTRAINT UQ_CihazModel UNIQUE (MarkaID, Model)
);

```

#### 15. Cihaz Markaları Tablosu

```

CREATE TABLE CihazMarkalari (
    MarkaID INT IDENTITY(1,1) PRIMARY KEY,
    Marka NVARCHAR(50) NOT NULL UNIQUE
);

```

#### 16. Cihazlar Tablosu:

```

CREATE TABLE Cihazlar (
    CihazID INT IDENTITY(1,1) PRIMARY KEY,
    MusteriID INT NOT NULL,
    AbonelikID INT NULL,
    CihazTipiID INT NULL,
    ModelID INT NULL,
    SeriNumarasi NVARCHAR(50) NOT NULL UNIQUE,
    VerilisTarihi DATE NOT NULL,
    Durum BIT NOT NULL CHECK (Durum IN (0, 1)),

```



```

        CONSTRAINT FK_Cihazlar_Musteriler FOREIGN KEY (MusteriID) REFERENCES
Musteriler(MusteriID),
        CONSTRAINT FK_Cihazlar_Abonelikler FOREIGN KEY (AbonelikID) REFERENCES
Abonelikler(AbonelikID),
        CONSTRAINT FK_Cihazlar_CihazTipi FOREIGN KEY (CihazTipiID) REFERENCES
CihazTipleri(CihazTipiID),
        CONSTRAINT FK_Cihazlar_CihazModel FOREIGN KEY (ModelID) REFERENCES
CihazModelleri(ModelID)
);

```

#### 17. Kampanyalar Tablosu:

```

CREATE TABLE Kampanyalar (
    KampanyaID INT IDENTITY(1,1) PRIMARY KEY,
    KampanyaAdi NVARCHAR(100) NOT NULL,
    BaslangicTarihi DATE NOT NULL,
    BitisTarihi DATE NOT NULL,
    CONSTRAINT CK_Kampanya_Bitis_Tarihi CHECK (BitisTarihi >= BaslangicTarihi),
    IndirimOrani DECIMAL(5, 2) NOT NULL CHECK (IndirimOrani >= 0 AND IndirimOrani <=
100),
    Aciklama NVARCHAR(MAX)
);

```

#### 18. Kampanya Paketleri Tablosu:

```

CREATE TABLE KampanyaPaketleri (
    KampanyaPaketID INT IDENTITY(1,1) PRIMARY KEY,
    KampanyaID INT NOT NULL,
    PaketID INT NOT NULL,
    EkAciklama NVARCHAR(255),
    CONSTRAINT FK_KampanyaPaketleri_Kampanyalar FOREIGN KEY (KampanyaID) REFERENCES
Kampanyalar(KampanyaID),
    CONSTRAINT FK_KampanyaPaketleri_Paketler FOREIGN KEY (PaketID) REFERENCES
Paketler(PaketID)
);

```

#### 19. Log Varlık Tipleri Tablosu:

```

CREATE TABLE LogVarlikTipleri (
    VarlikTipiID INT IDENTITY(1,1) PRIMARY KEY,
    VarlikTipi NVARCHAR(50) NOT NULL UNIQUE,
    Aciklama NVARCHAR(MAX)
);

```

#### 20. Log Tipleri Tablosu:

```

CREATE TABLE LogTipleri (
    LogTipiID INT IDENTITY(1,1) PRIMARY KEY,
    IslemTipi NVARCHAR(50) NOT NULL,
    VarlikTipiID INT NOT NULL,
    CONSTRAINT FK_LogTipleri_VarlikTipleri FOREIGN KEY (VarlikTipiID) REFERENCES
LogVarlikTipleri(VarlikTipiID)
);

```

#### 21. Sistem Loglari Tablosu:

```

CREATE TABLE SistemLoglari (
    LogID INT IDENTITY(1,1) PRIMARY KEY,
    VarlikID INT NOT NULL,
    LogTipiID INT NOT NULL,
    IslemTarihi DATETIME NOT NULL,
    Ayrntilar NVARCHAR(MAX),
    CONSTRAINT FK_SistemLoglari_LogTipleri FOREIGN KEY (LogTipiID) REFERENCES
LogTipleri(LogTipiID)
);

```

## 22. Bakım Tipleri Tablosu:

```
CREATE TABLE BakimTipleri (  
    BakimTipiID INT IDENTITY(1,1) PRIMARY KEY,  
    BakimTipiAdi NVARCHAR(50) NOT NULL,  
    Aciklama NVARCHAR(MAX)  
);
```

## 23. Bakım Durumları Tablosu:

```
CREATE TABLE BakimDurumlari (  
    DurumID INT IDENTITY(1,1) PRIMARY KEY,  
    DurumAdi NVARCHAR(50) NOT NULL UNIQUE,  
    Aciklama NVARCHAR(MAX)  
);
```

## 24. Bakım Kayıtları Tablosu:

```
CREATE TABLE BakimKayitlari (  
    BakimID INT IDENTITY(1,1) PRIMARY KEY,  
    PersonelID INT NOT NULL,  
    BakimTipiID INT NOT NULL,  
    BakimTarihi DATE NOT NULL,  
    Aciklama NVARCHAR(255),  
    DurumID INT NOT NULL,  
    CONSTRAINT FK_BakimKayitlari_Personel FOREIGN KEY (PersonelID) REFERENCES  
Personel(PersonelID),  
    CONSTRAINT FK_BakimKayitlari_BakimTipleri FOREIGN KEY (BakimTipiID) REFERENCES  
BakimTipleri(BakimTipiID),  
    CONSTRAINT FK_BakimKayitlari_Durum FOREIGN KEY (DurumID) REFERENCES  
BakimDurumlari(DurumID)  
);
```

## 3) Örnek Veri Ekleme

### 1. İletişim Bilgileri Tablosuna Veri Ekleme:

```
INSERT INTO IletisimBilgileri (Telefon, Email, Il, Ilce, Mahalle, Sokak, Apartman,  
KapiNo)  
VALUES  
( '5443692485', 'halilibro921@gmail.com', 'Denizli', 'Pamukkale', 'Beyazıt',  
'Kelmahmut', 'Portakal', '1'),  
( '5436965848', 'cagri_soysal34@hotmail.com', 'Istanbul', 'Sisli', 'Mecidiyekoy',  
'Buyukdere', 'Plaza', '10'),  
( '5593486922', 'yusufAydin@proton.me', 'Kayseri', 'Melikgazi', 'Aliler', 'Mustafa  
Kemal Pasa', 'Remax', '19'),  
  
( '5423691599', 'salim.teknik1@iss.com', 'Istanbul', 'Kadikoy', 'Caferaga',  
'karabagci', 'Balci', '14'),  
  
( '5448396502', 'kemal.saha1@iss.com', 'Istanbul', 'Besiktas', 'Levent', 'karasalci',  
'Salci', '22'),  
( '5503698751', 'melekyorden@proton.me', 'Istanbul', 'Bagcilar', 'Mecidiyekoy',  
'cemisgezek', 'Remax', '9'),  
( '5414141254', 'esrabagli@gmail.com', 'Istanbul', 'Uskudar', 'Altunizade', 'shibuya',  
'Elysium', '1');
```

### 2. Musteriler Tablosuna Veri Ekleme:

```
INSERT INTO Musteriler (Ad, Soyad, Iletisim_ID, KayitTarihi)  
VALUES  
( 'Halil Ibrahim', 'Yilmaz', 1, '2024-01-04'),  
( 'Cagri', 'Soysal', 2, '2024-01-03');
```

### 3. Paketler Tablosuna Veri Ekleme:

```
INSERT INTO Paketler (PaketAdi, Hiz, Kota, AylikUcret, Durum)
```

```
VALUES
('Fiber 50Mbps', 50, 100, 199.99, 1),
('Fiber 100Mbps', 100, 250, 299.99, 1),
('Fiber Ultra', 200, 1000, 499.99, 1);
```

#### 4. LogVarlıkTipleri Tablosuna Veri Ekleme:

```
-- Log Varlık Tipleri tablosuna veri ekleme
INSERT INTO LogVarlıkTipleri (VarlıkTipi, Aciklama)
VALUES
('Odeme', 'Ödeme işlemleri ile ilgili loglar'),
('Abonelik', 'Abonelik işlemleri ile ilgili loglar'),
('Musteri', 'Müşteri işlemleri ile ilgili loglar');
```

#### 5. LogTipleri Tablosuna Veri Ekleme:

```
-- LogTipleri tablosuna veri ekle
INSERT INTO LogTipleri (IslemTipi, VarlıkTipiID)
VALUES
('Odeme Yapildi', (SELECT VarlıkTipiID FROM LogVarlıkTipleri WHERE VarlıkTipi =
'Odeme')),
('Odeme iptal', (SELECT VarlıkTipiID FROM LogVarlıkTipleri WHERE VarlıkTipi =
'Odeme')),
('Abonelik Basladi', (SELECT VarlıkTipiID FROM LogVarlıkTipleri WHERE VarlıkTipi =
'Abonelik')),
('Abonelik Sonlandi', (SELECT VarlıkTipiID FROM LogVarlıkTipleri WHERE VarlıkTipi =
'Abonelik')),
('Musteri Eklendi', (SELECT VarlıkTipiID FROM LogVarlıkTipleri WHERE VarlıkTipi =
'Musteri')),
('Musteri Guncellendi', (SELECT VarlıkTipiID FROM LogVarlıkTipleri WHERE VarlıkTipi =
'Musteri')),
('Musteri Silindi', (SELECT VarlıkTipiID FROM LogVarlıkTipleri WHERE VarlıkTipi =
'Musteri')),

('Paket Degistirildi', (SELECT VarlıkTipiID FROM LogVarlıkTipleri WHERE VarlıkTipi =
'Abonelik'));
```

#### 6. OdemeYontemleri Tablosuna Veri Ekleme:

```
INSERT INTO OdemeYontemleri (OdemeYontemiAdi, Aciklama)
VALUES
('Kredi Kartı', 'Kredi kartları ile yapılan ödemeler.'),
('Banka Havalesi', 'Banka hesabı üzerinden yapılan para transferi ile ödeme.'),
('Mobil Ödeme', 'Cep telefonu numarası üzerinden yapılan ödeme işlemleri.');
```

#### 7. TeknikDestekDurumlari Tablosuna Veri Ekleme:

```
INSERT INTO TeknikDestekDurumlari (Durum, Aciklama)
VALUES
('Yeni', 'Henüz işleme alınmamış yeni destek talebi'),
('Beklemede', 'Değerlendirme aşamasında olan talepler'),
('Atandı', 'Teknik personele atanmış, çözüm sürecinde'),
('Müşteri Yanıtı Bekleniyor', 'Müşteriden ek bilgi veya onay bekleniyor'),
('Sahada', 'Teknik ekip müşteri lokasyonunda'),
('Test Aşamasında', 'Çözüm uygulandı, test ediliyor'),
('Çözüldü', 'Teknik destek talebi çözüme ulaştı'),
('Kapalı', 'Talep sonlandırıldı ve arşivlendi'),
('İptal Edildi', 'Müşteri talebi ile iptal edildi'),
('Tekrar Açıldı', 'Çözülen talep tekrar açıldı');
```

#### 8. Roller Tablosuna Veri Ekleme:

```
INSERT INTO Roller (RolAdi, Aciklama)
VALUES
('Teknik Destek Uzmanı', 'Müşteri teknik sorunlarını çözen personel'),
('Teknik Ekip Lideri', 'Teknik ekibi yöneten ve koordine eden personel'),
('Saha Teknisyeni', 'Saha kurulum ve onarım işlerini yapan personel'),
('Network Uzmanı', 'Ağ altyapısı ve optimizasyonundan sorumlu personel');
```

```
( 'Magaza Musteri Danismani', 'Magazada musterilerle ilgilenen personel'),  
( 'Magaza Muduru', 'Magaza yonetiminden sorumlu personel');
```

#### 9. Magazalar Tablosuna Veri Ekleme:

```
INSERT INTO Magazalar (MagazaAdi, Iletisim_ID, Aciklama)  
VALUES ('Kadıköy Teknik Merkez',  
(SELECT IletisimID FROM IletisimBilgileri WHERE Email = 'salim.teknik1@iss.com'),  
'Ana teknik destek merkezi');
```

#### 10. Personel Tablosuna Veri Ekleme:

```
INSERT INTO Personel (MagazaID, Ad, Soyad, Iletisim_ID, RolID, GorevBaslangicTarihi,  
Maas)  
SELECT  
(SELECT TOP 1 MagazaID FROM Magazalar),  
Ad,  
Soyad,  
IletisimID,  
RolID,  
'2024-01-05',  
Maas  
FROM (  
VALUES  
( 'Salim', 'Berke',  
(SELECT IletisimID FROM IletisimBilgileri WHERE Email =  
'salim.teknik1@iss.com'),  
(SELECT RolID FROM Roller WHERE RolAdi = 'Teknik Destek Uzmanı'),  
8500.00),  
( 'Kemal', 'Yonca',  
(SELECT IletisimID FROM IletisimBilgileri WHERE Email =  
'kemal.sahal@iss.com'),  
(SELECT RolID FROM Roller WHERE RolAdi = 'Saha Teknisyeni'),  
7500.00),  
( 'Melek', 'Yorden',  
(SELECT IletisimID FROM IletisimBilgileri WHERE Email =  
'melekyorden@proton.me'),  
(SELECT RolID FROM Roller WHERE RolAdi = 'Magaza Musteri Danismani'),  
6000.00),  
( 'Esra', 'Bagli',  
(SELECT IletisimID FROM IletisimBilgileri WHERE Email =  
'esrabagli@gmail.com'),  
(SELECT RolID FROM Roller WHERE RolAdi = 'Magaza Muduru'),  
10000.00)  
) AS PersonelVeri(Ad, Soyad, IletisimID, RolID, Maas);
```

#### 11. Abonelikler Tablosuna Veri Ekleme:

```
INSERT INTO Abonelikler (MusteriID, PaketID, BaslangicTarihi, BitisTarihi, Durum)  
VALUES  
(1, 1, '2024-01-04', NULL, 1),  
(2, 2, '2024-01-03', NULL, 1);
```

#### 12. Faturalar Tablosuna Veri Ekleme:

```
INSERT INTO Faturalar (AbonelikID, FaturaTarihi, SonOdemeTarihi, FaturaTutari, Durum)  
VALUES  
(1, '2024-01-04', '2024-02-04', 199.99, 0),  
(2, '2024-01-03', '2024-02-03', 299.99, 1);
```

#### 13. Ödemeler Tablosuna Veri Ekleme:

```
INSERT INTO Odemeler (AbonelikID, OdemeYontemiID, OdemeTarihi, OdemeMiktari)  
VALUES  
(2, 1, '2024-01-03', 299.99),  
(1, 2, '2024-01-15', 199.99);
```

#### 14. TeknikDestekTalepleri Tablosuna Veri Ekleme:

```
INSERT INTO TeknikDestekTalepleri (MusteriID, PersonelID, TalepTarihi, DurumID,
Aciklama, CozumTarihi)
VALUES
(1, 1, '2024-01-05', 1, 'İnternet bağlantısı kesiliyor', NULL),
(2, 2, '2024-01-06', 3, 'Modem arızası', NULL),
(1, 3, '2024-01-07', 7, 'Yavaş internet sorunu', '2024-01-07');
```

#### 15. SistemLoglari Tablosuna Veri Ekleme

```
INSERT INTO SistemLoglari (VarlikID, LogTipiID, IslemTarihi, Ayrıntılar)
VALUES
(1, 1, '2024-01-05 10:30:00', 'Yeni abonelik başlatıldı'),
(2, 3, '2024-01-05 11:45:00', 'Ödeme alındı'),
(1, 5, '2024-01-06 09:15:00', 'Müşteri bilgileri güncellendi');
```

#### 16. CihazTipleri Tablosuna Veri Ekleme:

```
INSERT INTO CihazTipleri (CihazTipiAdi, Aciklama)
VALUES
('Modem', 'Internet bağlantısı için modem cihazı'),
('Router', 'Kablosuz ağ yönlendiricisi');
```

#### 17. CihazMarkalari Tablosuna Veri Ekleme:

```
INSERT INTO CihazMarkalari (Marka)
VALUES
('TP-Link'),
('Huawei'),
('Zyxel'),
('Asus');
```

#### 18. CihazModelleri Tablosuna Veri Ekleme:

```
INSERT INTO CihazModelleri (MarkaID, Model)
VALUES
(1, 'Archer VR300'),
(2, 'HG658 V2'),
(3, 'VMG1312-T20B'),
(4, 'DSL-AX82U');
```

#### 19. Cihazlar Tablosuna Veri Ekleme:

```
INSERT INTO Cihazlar (MusteriID, AbonelikID, CihazTipiID, ModelID, SeriNumarasi,
VerilisTarihi, Durum)
VALUES
(2, 2, 1, 3, 'SN982564762145', '2024-01-05', 1);
```

#### 20. Kampanyalar Tablosuna Veri Ekleme:

```
INSERT INTO Kampanyalar (KampanyaAdi, BaslangicTarihi, BitisTarihi, IndirimOrani,
Aciklama)
VALUES
('Yeni Yıl Kampanyası', '2024-01-01', '2024-02-01', 25.00, '2025 yılına özel indirim kampanyası'),
('Öğrenci Kampanyası', '2024-02-24', '2024-06-30', 30.00, 'Öğrencilere özel indirim'),
('Yaz Fırsatı', '2024-06-01', '2024-08-31', 20.00, 'Yaz aylarına özel indirim kampanyası');
```

#### 21. Kampanya Paketleri Tablosuna Veri Ekleme:

```
INSERT INTO KampanyaPaketleri (KampanyaID, PaketID, EkAciklama)
VALUES
(1, 1, 'Fiber 50Mbps paketi için geçerli'),
(1, 2, 'Fiber 100Mbps paketi için geçerli'),
(2, 1, 'Öğrenciler için uygun paket'),
(3, 3, 'Ultra paket yaz indirimi');
```

#### 22. BakimTipleri Tablosuna Veri Ekleme:

```
INSERT INTO BakimTipleri (BakimTipiAdi, Aciklama)
VALUES
('Rutin Kontrol', 'Periyodik rutin altyapı kontrolleri'),
('Arıza Giderme', 'Arıza durumunda yapılan bakım'),
('Güncelleme', 'Sistem güncellemeleri ve yükseltmeleri');
```

#### 23. BakimDurumlari Tablosuna Veri Ekleme:

```
INSERT INTO BakimDurumlari (DurumAdi, Aciklama)
VALUES
('Planlandı', 'Bakım işlemi planlandı, henüz başlamadı'),
('Devam Ediyor', 'Bakım işlemi devam ediyor'),
('Tamamlandı', 'Bakım işlemi başarıyla tamamlandı'),
('İptal Edildi', 'Bakım işlemi iptal edildi');
```

#### 24. BakimKayitlari Tablosuna Veri Ekleme:

```
INSERT INTO BakimKayitlari (PersonelID, BakimTipiID, BakimTarihi, Aciklama, DurumID)
VALUES
(1, 1, '2024-01-05', 'Periyodik rutin bakım yapıldı', 3),
(2, 2, '2024-01-05', 'Fiber kablo arızası giderildi', 3),
(1, 3, '2024-01-06', 'Sistem güncellemesi yapılacak', 1);
```

## 4) Saklı Yordam (Stored Procedure) Örnekleri

### 1. Yeni Abonelik Oluşturma

=> Verilen MusteriID, PaketID ve BaslangicTarihi'ni kullanarak yeni bir abonelik başlatır ve ilk faturayı keser. Eğer bir hata oluşursa yapılan işlemleri geri alır.

```
-- Yeni Abonelik için saklı yordam (stored procedure) ve transaction
CREATE PROCEDURE sp_YeniAbonelikOlustur
    @MusteriID INT,
    @PaketID INT,
    @BaslangicTarihi DATE
AS
BEGIN
    BEGIN TRY
        BEGIN TRANSACTION;

        -- Musteri var mı kontrol et. Yoksa hata ver.
        IF NOT EXISTS (SELECT 1 FROM Musteriler WHERE MusteriID = @MusteriID)
        BEGIN
            THROW 51000, 'Müşteri bulunamadı.', 1;
        END

        -- Paket var mı ve varsa aktif mi kontrol et. Yoksa hata ver.
        IF NOT EXISTS (SELECT 1 FROM Paketler WHERE PaketID = @PaketID AND Durum = 1)
        BEGIN
            THROW 51001, 'Paket bulunamadı veya aktif değil.', 1;
        END

        -- Yeni abonelik Oluştur
        INSERT INTO Abonelikler (MusteriID, PaketID, BaslangicTarihi, Durum)
        VALUES (@MusteriID, @PaketID, @BaslangicTarihi, 1);

        -- AbonelikId'yi al
        DECLARE @AbonelikID INT = SCOPE_IDENTITY();

        -- İlk Faturayı oluştur
```

```

        DECLARE @FaturaTutari DECIMAL(10,2);
        SELECT @FaturaTutari = AylikUcret
        FROM Paketler
        WHERE PaketID = @PaketID;

        INSERT INTO Faturalar (AbonelikID, FaturaTarihi, SonOdemeTarihi, FaturaTutari,
Durum)
        VALUES (@AbonelikID, @BaslangicTarihi, DATEADD(DAY, 30, @BaslangicTarihi),
@FaturaTutari, 0);

        COMMIT TRANSACTION;
    END TRY
    BEGIN CATCH
        ROLLBACK TRANSACTION;
        THROW 51002, 'Hata oluřtu, iřlemler geri alındı', 1;
    END CATCH
END;

```

## 2. Abonelik iptal Etme:

=> Verilen AbonelikID'ye sahip abonelięi ve son faturayı iptal eder. Ardından verilen IptalNedini'ni kullanarak yeni bir log oluřturur. Bu iřlemler sırasında herhangi bir hata oluřursa yapılan tüm iřlemler geri alınır.

```

-- Abonelik iptali için saklı yordam (stored procedure) ve transaction
CREATE PROCEDURE sp_AbonelikIptal
    @AbonelikID INT,
    @IptalNedeni NVARCHAR(MAX)
AS
BEGIN
    BEGIN TRY
        BEGIN TRANSACTION;

        -- Abonelik mevcut mu veya aktif mi kontrol et.
        IF NOT EXISTS (SELECT 1 FROM Abonelikler WHERE AbonelikID = @AbonelikID AND
Durum = 1)
        BEGIN
            THROW 51005, 'Abonelik bulunamadı veya zaten iptal edilmiş.', 1;
        END

        -- Loga yüklemek için abonelik detaylarını al
        DECLARE @MusteriID INT;
        SELECT @MusteriID = MusteriID
        FROM Abonelikler
        WHERE AbonelikID = @AbonelikID;

        -- Abonelięin bitiř tarihi ve durumunu güncelle
        UPDATE Abonelikler
        SET
            Durum = 0,
            BitisTarihi = GETDATE()
        WHERE AbonelikID = @AbonelikID;

        -- Ödenmemiş faturaları iptal et
        UPDATE Faturalar
        SET Durum = 1
        WHERE AbonelikID = @AbonelikID
        AND Durum = 0
        AND FaturaTarihi <= GETDATE();

        -- Bu abonelikle ilgili cihazları deaktif et
        UPDATE Cihazlar
        SET

```

```

        Durum = 0,
        AbonelikID = NULL
    WHERE AbonelikID = @AbonelikID;

-- Log için gerekli LogTipiID'yi bul
    DECLARE @LogTipiID INT;
    SELECT @LogTipiID = lt.LogTipiID
    FROM LogTipleri lt
    INNER JOIN LogVarlikTipleri lvt ON lt.VarlikTipiID = lvt.VarlikTipiID
    WHERE lvt.VarlikTipi = 'Abonelik'
    AND lt.IslemTipi = 'Abonelik Sonlandi';

-- Logla
    INSERT INTO SistemLoglari (VarlikID, LogTipiID, IslemTarihi, Ayrntilar)
    VALUES (
        @AbonelikID,
        @LogTipiID,
        GETDATE(),
        CONCAT('Abonelik iptal edildi. Sebep: ', @IptalNedeni)
    );

    COMMIT TRANSACTION;
END TRY
BEGIN CATCH
    ROLLBACK TRANSACTION;
    THROW 51006, 'Abonelik iptali sırasında hata oluştu.', 1;
END CATCH
END;

```

### 3. Teknik Destek Talebi Oluşturma:

=> Verilen MusteriID'yi ve Acıklama'yı kullanarak yeni bir destek talebi oluşturur. Bu işlem sırasında bir hata oluşursa yapılan bütün işlemler geri alınır.

```

-- Teknik destek talebi olusturmak için saklı yordam (stored procedure) ve transaction.
CREATE PROCEDURE sp_TeknikDestekTalebiOlustur
    @MusteriID INT,
    @Aciklama NVARCHAR(MAX)
AS
BEGIN
    DECLARE @YeniTalepDurumID INT;

    BEGIN TRY
        -- "Yeni" durumu için DestekDurumID'yi bul
        SELECT @YeniTalepDurumID = DestekDurumID
        FROM TeknikDestekDurumlari
        WHERE Durum = 'Yeni';

        -- Eğer öyle bi durum yoksa hata ver
        IF @YeniTalepDurumID IS NULL
        BEGIN
            THROW 51007, 'Sistem hatası: "Yeni" durum tanımı bulunamadı.', 1;
        END

        BEGIN TRANSACTION;
        -- Müşterinin varlığını ve aktif bir aboneliğinin olup olmadığını kontrol et.
        IF NOT EXISTS (
            SELECT 1
            FROM Musteriler m
            JOIN Abonelikler a ON m.MusteriID = a.MusteriID
            WHERE m.MusteriID = @MusteriID AND a.Durum = 1

```



```
)
BEGIN
    THROW 51003, 'Müşteri bulunamadı veya aktif aboneliği yok.', 1;
END
```

```
-- Destek talebi oluştur.
INSERT INTO TeknikDestekTalepleri (
    MusteriID,
    TalepTarihi,
    DurumID,
    Aciklama
)
VALUES (
    @MusteriID,
    GETDATE(),
    @YeniTalepDurumID,
    @Aciklama
);
```

```
COMMIT TRANSACTION;
END TRY
BEGIN CATCH
    ROLLBACK TRANSACTION;
    THROW 51004, 'Hata oluştu, işlemler geri alındı', 1;
    THROW;
END CATCH
```

```
END;
```

#### 4. Destek Talebine Personele Atama

=> Mevcuttaki bir destek talebine bir personel atar. Eğer işlemler sırasında bir hata olursa yapılan işlemleri geri alır ve kodu bitirir.

```
-- 4. Teknik Destek Talebine Personel Atama
CREATE PROCEDURE sp_TeknikDestekAtama
    @DestekTalepID INT,
    @PersonelID INT
AS
BEGIN
    BEGIN TRY
        BEGIN TRANSACTION;
        -- Verilen destek talebi mevcut mu veya başka bir personel atanmış mı?
        IF NOT EXISTS (
            SELECT 1
            FROM TeknikDestekTalepleri
            WHERE DestekTalepID = @DestekTalepID
            AND PersonelID IS NULL
        )
        BEGIN
            THROW 51009, 'Destek talebi bulunamadı veya daha önce atanmış', 1;
        END

        -- Personel mevcut mu?
        IF NOT EXISTS (
            SELECT 1
            FROM Personel
            WHERE PersonelID = @PersonelID
        )
        BEGIN
            THROW 51010, 'Personel bulunamadı', 1;
        END
    END TRY
    BEGIN CATCH
        ROLLBACK TRANSACTION;
        THROW 51011, 'Hata oluştu, işlemler geri alındı', 1;
    END CATCH
END
```

```
END
```

```
-- 'Atandı' durumu için DestekDurumID'yi bul  
DECLARE @AtandiDurumID INT;  
SELECT @AtandiDurumID = DestekDurumID  
FROM TeknikDestekDurumlari  
WHERE Durum = 'Atandı';
```

```
-- Destek talebini güncelle  
UPDATE TeknikDestekTalepleri  
SET  
    PersonelID = @PersonelID,  
    DurumID = @AtandiDurumID  
WHERE DestekTalepID = @DestekTalepID;
```

```
COMMIT TRANSACTION;  
END TRY  
BEGIN CATCH  
    ROLLBACK TRANSACTION;  
    THROW 51011, 'Atama yaparken hata oluştu. Rollback yapıldı', 1;  
    THROW;  
END CATCH  
END;
```

## 5) Triggerlar

### 1) Ödeme Yapıldığında Fatura Güncelleme:

=> Bir ödeme yapıldığında ilgili faturanın durumu 1 yani ödendi olarak güncellenir. Ardından log olarak SistemLoglari tablosuna eklenir.

```
-- Ödeme yapıldığında faturalarda ödeme durumu güncelleyen trigger.
```

```
CREATE TRIGGER trg_OdemeKontrol  
ON Odemeler  
AFTER INSERT  
AS  
BEGIN  
    BEGIN TRY  
        -- Ödeme yapıldığında ilgili fatura durumunu güncelle  
        UPDATE f  
        SET f.Durum = 1  
        FROM Faturalar f  
        INNER JOIN inserted i ON f.AbonelikID = i.AbonelikID  
        WHERE f.FaturaTutari = i.OdemeMiktari AND f.Durum = 0;
```

```
-- Ödeme logu için doğru logTipiID'yi bul  
DECLARE @LogTipiID INT;  
SELECT @LogTipiID = lt.LogTipiID  
FROM LogTipleri lt  
INNER JOIN LogVarlikTipleri lvt ON lt.VarlikTipiID = lvt.VarlikTipiID  
WHERE lvt.VarlikTipi = 'Odeme'  
AND lt.IslemTipi = 'Odeme Yapildi';
```

```
-- Ödemeyi sistem loglarına ekle  
INSERT INTO SistemLoglari (VarlikID, LogTipiID, IslemTarihi, Ayrıntılar)  
SELECT  
    i.OdemeID,  
    @LogTipiID,  
    GETDATE(),  
    'Ödeme yapıldı ve fatura durumu güncellendi'  
FROM inserted i;  
END TRY
```

```

BEGIN CATCH
    THROW;
END CATCH
END;

```

## 2) Müşteri Bilgisi Güncellendiğinde log ekleme

=> Müşterinin herhangi bir bilgisi değiştiğinde SistemLoglarına bir log eklenir.

```

-- Müşteri güncellemelerinde log ekleme
CREATE OR ALTER TRIGGER trg_MusteriDegisiklikLog
ON Musteriler
AFTER UPDATE
AS
BEGIN
    DECLARE @LogTipiID INT;

    -- Müşteri güncelleme operasyonu için doğru logTipiID'yi bul
    SELECT @LogTipiID = LogTipiID
    FROM LogTipleri lt
    JOIN LogVarlikTipleri lvt ON lt.VarlikTipiID = lvt.VarlikTipiID
    WHERE lvt.VarlikTipi = 'Musteri'
    AND lt.IslemTipi = 'Musteri Guncellendi';

    INSERT INTO SistemLoglari (VarlikID, LogTipiID, IslemTarihi, Ayrıntılar)
    SELECT
        i.MusteriID,
        @LogTipiID,
        GETDATE(),
        'Müşteri bilgileri güncellendi: ' +
        CASE
            WHEN UPDATE(Ad) OR UPDATE(Soyad)
            THEN 'Kişisel bilgiler değiştirildi (Ad/Soyad)'
            WHEN UPDATE(Iletisim_ID)
            THEN 'İletişim bilgileri referansı değiştirildi'
            ELSE 'Diğer bilgiler değiştirildi'
        END
    FROM inserted i;
END;

```

## 3. Aktif Aboneliği Silmeyi Engelleyen Trigger

=> Aktif durumda olan bir aboneliği silmeyi engeller.

```

-- Aktif aboneliği silmeyi engelleyen trigger
CREATE TRIGGER trg_AktifAbonelikSilmeEngelleme
ON Abonelikler
INSTEAD OF DELETE
AS
BEGIN
    IF EXISTS (
        SELECT 1
        FROM deleted
        WHERE Durum = 1
    )
    BEGIN
        THROW 51008, 'Aktif olan bir abonelik silinemez. Lütfen önce deaktive edin.',
        1;
        RETURN;
    END

    DELETE a
    FROM Abonelikler a
    INNER JOIN deleted d ON a.AbonelikID = d.AbonelikID;

```

END;

## 6) Transaction

### 1) Paket Değiştirme:

=> Verilen AbonelikID'de kullanılan paketi YeniPaketID ile değiştirir. İşlemler sırasında bir hata olması durumunda yapılan işlemler geri alınır.

```
-- Paket değiştirme prosedürü ve transaction.
CREATE PROCEDURE sp_PaketDegistir
    @AbonelikID INT,
    @YeniPaketID INT
AS
BEGIN
    DECLARE @EskiPaketID INT;
    DECLARE @MusteriID INT;
    DECLARE @LogTipiID INT;
    DECLARE @VarlikTipiID INT;

    BEGIN TRY
        BEGIN TRANSACTION;

        -- Şimdi kullanılan paketi ve müşteri bilgilerini al
        SELECT @EskiPaketID = PaketID, @MusteriID = MusteriID
        FROM Abonelikler
        WHERE AbonelikID = @AbonelikID;

        IF @EskiPaketID IS NULL
        BEGIN
            THROW 51004, 'Abonelik bulunamadı.', 1;
        END

        -- Aboneliği yeni paket ile güncelle
        UPDATE Abonelikler
        SET PaketID = @YeniPaketID
        WHERE AbonelikID = @AbonelikID;

        -- Yeni paket için fatura oluştur
        DECLARE @YeniUcret DECIMAL(10,2);
        SELECT @YeniUcret = AylikUcret
        FROM Paketler
        WHERE PaketID = @YeniPaketID;

        INSERT INTO Faturalar (AbonelikID, FaturaTarihi, SonOdemeTarihi, FaturaTutari,
Durum)
        VALUES (@AbonelikID, GETDATE(), DATEADD(DAY, 30, GETDATE()), @YeniUcret, 0);

        -- LogTipiID'yi dinamik olarak belirle
        SELECT @VarlikTipiID = VarlikTipiID
        FROM LogVarlikTipleri
        WHERE VarlikTipi = 'Abonelik'; -- Abonelik işlemleri için uygun varlık tipi

        SELECT @LogTipiID = LogTipiID
        FROM LogTipleri
        WHERE IslemTipi = 'Paket Degistirildi' AND VarlikTipiID = @VarlikTipiID;

        -- Değişikliği loga kaydet
        INSERT INTO SistemLoglari (VarlikID, LogTipiID, IslemTarihi, Ayrıntılar)
        VALUES (
            @AbonelikID,
            @LogTipiID,
```

```

        GETDATE(),
        CONCAT('Paket değiştirildi: Eski Paket ID = ', @EskiPaketID, ', Yeni Paket
ID = ', @YeniPaketID)
    );

    COMMIT TRANSACTION;
END TRY
BEGIN CATCH
    ROLLBACK TRANSACTION;
    THROW;
END CATCH
END;

```

## 7) Triggerları ve Prosedürleri Deneme

1 numaralı kullanıcı ve 1 numaralı paketi kullanıp yeni bir abonelik ve fatura oluşturur:

```
EXEC sp_YeniAbonelikOlustur @MusteriID = 1, @PaketID = 1, @BaslangicTarihi = '2024-01-04';
```

- 2 numaralı müşteri internet yavaş açıklamalı bir destek talebi oluşturuyor.

```
EXEC sp_TeknikDestekTalebiOlustur @MusteriID = 2, @Aciklama = 'Internet bağlantısı yavaş';
```

- 1 numaralı abonenin paketini 2 numaralı paketle değiştirir. Yeni bir log ve fatura oluşturur:

```
EXEC sp_PaketDegistir @AbonelikID = 1, @YeniPaketID = 2;
```

- 1 numaralı abonenin 1 numaralı ödemeye yöntemiyle faturasını öde. İlgili fatura durumu, ödemeler tablosu, sistem logları güncellenir. (Trigger)

```
INSERT INTO Odemeler (AbonelikID, OdemeYontemiID, OdemeTarihi, OdemeMiktari)
VALUES (1, 1, GETDATE(), 299.99);
```

- 1 numaralı aboneliği iptal et. Log oluştur, faturaları iptal et, abonelik durumunu değiştir, cihaz sahibini null yap.

```
EXEC sp_AbonelikIptal @AbonelikID = 1, @IptalNedeni = 'Memnun değil';
```

- Eğer olmayan bir aboneliği iptal etmeye kalkarsak

```
Msg 51006, Level 16, State 1, Procedure sp_AbonelikIptal, Line 64 [Batch Start Line 0]
Abonelik iptali sırasında hata oluştu."
```

diye bir hata alırız. İşlem iptal edilir (Rollback).

- Eğer aktif bir aboneliği silmeye kalkışırsak

```
Msg 51008, Level 16, State 1, Procedure trg_AktifAbonelikSilmeEngelleme, Line 13 [Batch Start Line 0]
```

Aktif olan bir abonelik silinemez. Lütfen önce deaktive edin  
Hatasını alırız.

- 3 numaralı destek talebine 1 numaralı personeli atama.

```
EXEC sp_TeknikDestekAtama @DestekTalepID = 3, @PersonelID = 1;
```